HAL open science

Designing techniques for systemic impact: lessons from C-K theory and matroid structures

Pascal Le Masson, Armand Hatchuel, Olga Kokshagina, Benoit Weil

To cite this version:

Pascal Le Masson, Armand Hatchuel, Olga Kokshagina, Benoit Weil. Designing techniques for systemic impact: lessons from C-K theory and matroid structures. Research in Engineering Design, 2016, 28, p. 275-298. 10.1007/s00163-016-0241-4 . hal-01389630

HAL Id: hal-01389630

https://hal.science/hal-01389630

Submitted on 28 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DESIGNING TECHNIQUES FOR SYSTEMIC IMPACT - LESSONS FROM C-K THEORY AND MATROID STRUCTURES

Authors:
Pascal Le Masson, Armand Hatchuel, Olga Kokshagina, Benoit Weil
MINES Paristech – PSL Research University, CGS-i3 – CNRS, UMR 9217.
Corresponding author: pascal.le_masson@mines-paristech.fr - tel: +33 1 40 51 92 21

Abstract

As underlined in Arthur's book "the nature of technology" we are very knowledgeable on the design of objects, services or technical systems, but we don't know much on the dynamics of technologies. Still contemporary innovation often consists in designing techniques with systemic impact. They are *pervasive*– both invasive and perturbing-, they recompose the family of techniques. Can we model the impact and the design of such techniques? More specifically: how can one design generic technology, ie a single technology that provokes a complete reordering of families of techniques?
Recent advances in design theories open new possibilities to answer these questions. In this paper we use C-K design theory and a matroid-based model of the set of techniques to propose a new model (C-K/Ma) of the dynamics of techniques, accounting for the design of generic technologies. We show:

- F1: C-K/Ma offers a computational model for designing a technique with systemic impact. This model sheds new light on the logics of combination in design – the model helps to identify four logics of "combinations", from non-generative combinatorics to generative one.
- F2: C-K/Ma accounts for some phenomena associated to generic technology design.
- F3: C-K/Ma offers an efficient guide for the design of technologies with systemic impact, based on generativity and genericity criteria

1 INTRODUCTION – DESIGNING FOR SYSTEMIC IMPACT?

In his book "the nature of technology"(Arthur 2009), W. Brian Arthur explains that looking for "some common logic that would structure technology and determine its ways and progress" he couldn't find it. Is this claim exaggerated in the light of the literature in the engineering design community? Not that much: we are very knowledgeable on the design of objects, services or technical systems. We are used to relying on techniques or building blocks to design them. But what do we know on the design of these "techniques" or building blocks? What do we know about the dynamics of techniques?
This question is all the more relevant today that the dynamics itself might be strongly changing. Let's consider some famous technologies and their impact: software in mechanical systems (aeronautics, mechatronics), semiconductors in a large variety of systems, additive processes (or 3D printing) in industrial processes, etc.All these technologies have a "systemic" impact, they are *pervasive* – both invasive and perturbing-, they recompose large parts of the set of existing techniques. Of course we know of some technologies that completely changed the technical environment of their time – "steam engine", "electricity"… One even associated each industrial revolution to a handful of such techniques. Today this process of *reorganizing* the family of techniques might be much more frequent. And it is not sure that we understand it: what is the origin of such pervasive techniques? How can we model their impact - rearrangements, "creative destruction", re-ordering? Are their design strategies to design the techniques and their impact? And even more specifically: how can one design generic technology, ie a single technology that provokes a complete reordering of families of techniques?
We tend to think about these dynamics of techniques in terms of "combination" and "assembly", maybe stuck in the "mechanical" paradigm. Or we rely on evolutionary models, with random emergence and "natural" selection. Some authors have tried to identify laws of the evolution of techniques. These proposals contributed to enlighten some facets of the dynamics of techniques but also raised difficult questions: why two independent domains of techniques become suddenly combinable? Can we differentiate between a "local" combination and a more "generic" one? Why

some techniques are just locally solving a problem while other might lead to generate entire lineages of descendants? how could we account for the design of a technique with an intentional systemic impact? Or more generally: we today understand how an new “individual” emerges in the order of techniques (the logic of “indivuation” analyzed by (Simondon 1958)) – or we understand how the order of techniques evolves –and occasionally gets stuck (see evolutionary models or the “blocked technical systems” of (Gilles 1986)). But how do we model and design the new individual that would be able to reorganize the whole system, ie a pervasive technique, or a generic technique?
Note that these questions raise a critical debate for design and design theory: on the one hand we tend to look at design (designed artifacts or techniques) as “combinations” of techniques – but on the other hand, combinatorics models don’t account for generativity in design – hence an issue for design theory: how can one understand and model a “generative combinatorics”?
Recent advances in design theories open new possibilities to answer these questions. In this paper we use C-K design theory and a matroid-based model of the set of techniques to propose a new model (C-K/Ma) of the dynamics of techniques, accounting for the design of generic technologies. We show that:

- F1: C-K/Ma offers a computational model of the process of designing a technique with systemic impact.
- F2: C-K/Ma offers an efficient model to account for some phenomena associated to generic technology design.
- F3: C-K/Ma offers a guide for the design of technologies with systemic impact, based on generativity and genericity criteria

2 RESEARCH QUESTIONS – INTERDEPENDENCES AND THE DESIGN OF THE SYSTEM OF TECHNIQUES

2.1 Endogenous dynamics of technologies : the design of interdependencies

In the multiple approaches on the dynamics of technologies, we can distinguish two main trends.
A first stream of works tends to consider that the dynamics of technologies is based on the invention of new functional means. Arthur’s book provide a synthesis of these approaches (Arthur 2009). Technologies are functional building blocks. The scientific study of phenomena regularly provides new building blocks. The functional building blocks are “*combined*” into artefacts. Combinations are selected – for instance by the markets. Hence the dynamics of technologies is controlled by market and science. This kind of model is more or less implicitly used in many evolutionary economics works(Dosi et al. 1988; Saviotti and Metcalfe 1991). This approach is “exogenous”: the dynamics of technologies mainly relies on exogenous forces – market and science. In this perspective, one tends to define a technique as a “functional” building block (eg Arthur considers that a technique is “a means to fulfil a human needs or purpose”), without considering the “combinations” issues. To be a bit caricatural: in this first approach, one tends to consider techniques as Lego blocks and one invention is the design of new kind of block. In the world of Lego, each new Lego block is designed to be compatible with all previously known blocks. Similarly, in the functional approach, the “combinative capability” of a new technique is implicitly considered as very high. This approach has some limits when it comes to situations where techniques are only partially compatible with each other.
As a consequence this approach tends to neglect the issue of the design and evolution of the combinative capabilities of techniques. In particular, in this perspective, it is more difficult to analyse the emergence of techniques whose main property is precisely the capacity to assure the compatibility *between* existing techniques. Let’s mention two contrasted cases: 1- when Watt and Boulton invents the cinematic for a rotary steam engine, they actually “just” make compatible the steam engine and the world of machine tools. We could say that this technique has a genericity property: it increases the genericity of steam angine (now compatible with new applications, beyond water puping in mines) and it increases the genericity of machine tools (now working beyond the limits of the usual energy source, hydraulic energy provided by rivers). 2- simpler illustrative case: the “swiss army knife” relies on the technique that enable to relate the different “tools” of the swiss knife – the articulation technique that enables to combine, for instance, a knife and a bottle opener.

Hence another, complementary perspective on technique: several authors, particularly in engineering design,have underlined that for one given set of functions, there are different ways –different combinations- to address it, and these different combinations don't have the same value. In particular, Design Structure Matrices (Ulrich and Eppinger 2008), or Modularity (Baldwin and Clark 2000), or Aximatic Design (Suh 2001) lead to distinguish between technological systems, although these systems seem equivalent from a functional point of view: systems with less interdependencies (DSM), with modularity (Baldwin's Design Rules) or with independences (first axiom of Axioamtic Design) are "better" – they are said to be more robust, easier to realise, or they enable a large variety of alternatives,… In this second stream of work, one has to consider the technique in the set of all existing techniques and the structure of this set. How should one describe this structure of the set of techniques? Several authors have noticed that the logic of "combination" is too fuzzy to describe the structures of technical systems. Precisely, Bertrand Gilles explains that, more than combinations, there are *interdependences*, "that within some limits, as a very general rule, all techniques are, to various degrees, depending on one another, so that there should be some coherence between them" (p. 19) (Gilles 1986). And consequently Bertrand Gilles defined a "technical system" as "the coherence, at different levels, of all the technical structures, of all the technological sets and ways " (p. 19). In this approach, the technique is not limited to its functional role but is also characterized by its *interdependences* with all other techniques (and subsets of techniques). And the authors are looking for "principles" to have an optimal form of interdependence between techniques. As a consequence, techniques "evolve" not only because of functional challenges and/or new scientific phenomena but also because of expected interdependences between techniques. In this second approach, market and science are not the only engines in the dynamics of technological system: the techniques themselves have their own, *endogenous dynamics*.
Studying the dynamics of techniques becomes studying the *evolutions in the interdependences in the set of techniques*. Engineering design literature has already studied how, in certain cases, the creation of specific interdependences have a *systemic impact:*enabling "modularity" or "diagonal matrices" or providing a new "common core" (Gawer 2009; Kokshagina et al. 2013a), some techniques open access to a large variety of configurations – they enable a logic of platform, of modularity; if they help to follow the first Suh Axiomatic axiom, then they enable to cover a larger range of functional combinations…This systemic impact can be characterized by criteria like generativity and genericity: for some authors, one should analyse the *generativity* of the design in a technical system, ie the new design paths that are opened by the new technique (Le Masson and Weil 2013; Hatchuel et al. 2011)); other authors have underlined that one single technique can immediately enable a large set of products following a pure "combinative" logic, ie with only very limited efforts, and hence the authors insist on the *genericity* of this technique(Kokshagina et al. 2013a)(Bresnahan and Trajtenberg 1995).For instance the rotary steam engine is generic because it suddenly enables new combinations of (existing) steam engines with every kinds of (existing) machines; it is also generative because it will further enable the design of completely new mobility systems like locomotives.

This quick overview of works on the dynamics of technologies provides us with clear advances:
1- Modelling techniques: understanding the design of techniquerequires to adopt *a modelof technique* in which a technique is characterized both by its functional input *and* the interdependences that this technique has with the set of techniques.
2- **A research issue:**The dynamics of techniques is precisely the evolution in the functions and in the interdependences. Research considers that these evolutions can be more or less "intentional". For instance Suh Axiomatic Design is normative – it prescribes that the set of techniques should evolve to ensure independences in systems; Simondonian theory of "hypertely" (Simondon 1958) is more descriptive and states that some techniques tend to increase interdependences. Hence, very generally, our issue:

- $\mathbf{Q_0}$*: to understand the "endogenous" dynamics of technology,* we need *to model design strategies with systemic impact* ($\mathbf{Q_0}$).

3- More precisely, **a first research question**: this requires to model the impact of the *interdependences* on design, in particular the effect of interdependences on generativity and genericity. Hence a more specific research question:

- **Q_1**: Can we find a *model thataccounts rigorously for "systemic impact",* and in particular for such notions as genericity, generativity, independence and dependence - can we *reach a computational and quantitative approach of these notions*?

4- And **a second research question**: understanding the endogenous dynamics of techniques requires*to model the impact of the design of techniques on the interdependences themselves*– ie how design creates specific interdependences. Hence our second research question:

- **Q_2**: How does such a model predictthe emergence and possibility of *strategies for the design of techniques with systemic impact (pervasive techniques)*? In particular, how can one model the *design of generic technologies*?

2.2 Design theory and knowledge structure

Recent advances in design theory provide us with elements to account for the design of techniques with "systemic impact". We can distinguish two different contributions: 1- how can we characterize the systemic impact with design theories – ie how do design theories account for *the impact of certain "systems of knowledge" on design possibilities*? (contribution to Q1); and: 2- how do design theories help *to model the design of specific knowledge structures*? (contribution to Q2).

1- Contribution to Q_1: design theories account largely for the impact of certain knowledge structures on the design of new artefacts. Historical analysis of Design Theories studied the effect of knowledge structure on the generative power in different cases (parametric design, systematic design) (Le Masson and Weil 2013). General Design Theory (Yoshikawa 1981; Reich 1995) shows how a knowledge structure with an Hausdorff measure (distinction criteria) warranties the design of any functional combination; Coupled Design Process (Braha and Reich 2003) shows that more generally, the set of functional combinations that can be reached depends on the "closure" of technologies, ie their neighbourhood of alternative technologies; in Axiomatic Design, when the DPs and FRs meet the first axiom, then larger ranges of functional requirements can be easily reached ; in Infused Design (Shai and Reich 2004a, b), design capacity increases with the rules in one domain and the laws linking different domains; in C-K theory the generative power comes from "holes" in the knowledge structure (Hatchuel et al. 2013); in Forcing, the knowledge base has to meet the "splitting condition" to enable the design of a set that is different from all already known sets (Hatchuel et al. 2013; Le Masson et al. 2013). Moreover, relying on K-reordering, a critical operation in Forcing and in C-K theory, one new design might also generate all the "combinations" of this new design with past "technologies" already known in K, hence a "generic" power linked to K-structure (Kokshagina et al. 2013a).
This review shows that *given a structure of dependences and independences in knowledge, Design Theories help to predict the impact of this structure on design capacities*. This impact can be characterized by theincrease (or decrease!) in the capacity to design further original design (*generativity*) and in the capacity to obtain a large set of artefacts by combination of the newly designed technique with all previously known techniques (*genericity*). *However they don't offer yet a computational and quantitative approach of generativity, genericity, dependence and independence.*

2- Contribution to Q_2: how do design theories model the design of a new "technology" characterized by its systemic impact? Or: how to design a specific "knowledge structure"? Actually design theories tend to favour the design of artefacts, given a certain knowledge structure; but they barely address the issue of designing a specific knowledge structure. Some insights are given in (Kokshagina et al. 2013a), based on a study of algebraic extensions with C-K theory: the authors show that an *algebraic* extension begins with a concept and, depending on the concept, it is possible to generate fields with specific structures – in particular a specific field size. This study was actually made possible by one key property of contemporary design theories: they don't depend on specific objects or domains – knowledge is a "free parameter". In this paper(Kokshagina et al. 2013a), the authors studied the design process associated to a knowledge structure reduced to a mathematical field and, hence, the generation of new fields.
Hence, even if design theory did not focus until now on the design of specific knowledge structures, design theory is a favourable framework to study this question. *In this paper, we willuse design theory to analyse design strategies applied on knowledge structures.*

2.3 Knowledge structures characterized (only) by independences: why relying on matroids

What is the relevant knowledge structure to study design strategies with systemic impact? As seen above, we expect that we are able to *characterize this knowledge structure by dependences and interdependences*, and we expect that the knowledge structure is such that generativity and genericity can be computed.

Let's take an example: in an axiomatic design problem (a non-diagonal system, ie a system with linear interdependences) we can use a design theory (like C-K design theory) to design a system that meets the first Axiomatic Design axiom, hence is now diagonal (see figure 1 below) – in this case we take as knowledge base the initial non diagonal system with its interdependences; the design process transforms the initial knowledge structure into a new knowledge structure with new independences (in this case the knowledge base is characterized by specific interdependences in linear algebra). We could also take as knowledge base a functional graph (with graphical relations between functions) and design a graph with less interdependences. We use design theory to change the dependence relations in a graph. Yet, in these examples, we are actually only interested by the *evolution of dependence relations* – be they based on linear algebra or graphs. Hence our analysis would be more general if applied to a knowledge structure that is only characterized by the interdependences between the elements – whatever the deep nature of the relation (graph or linear algebra). By chance mathematicians have already studied such a strange object in great detail: they call it *matroid*.

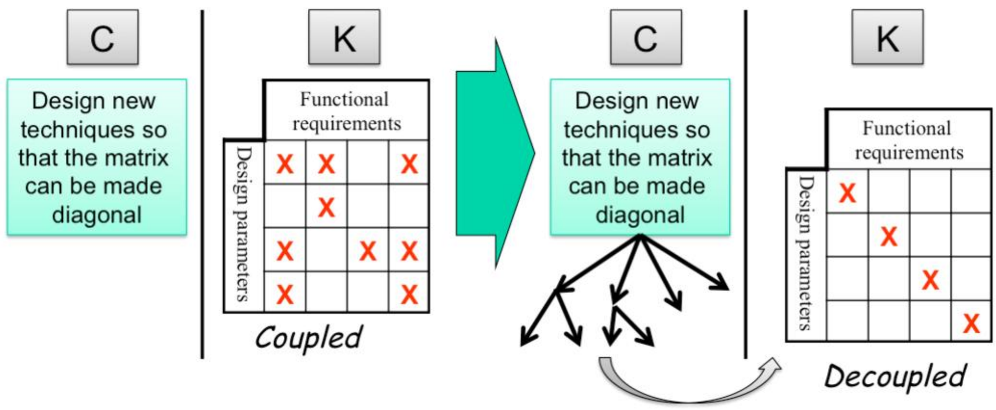


Figure 1: Designing a diagonalized matrix – left hand-side: before design; right hand-side: after design, the knowledge structure has changed.

Matroid structures were introduced by Whitney, in the 1930s (Whitney 1935), to capture abstractly the essence of (linear) dependence. Whitney explains his project as follows: "let C_1, C_2,… C_n be the columns of a matrix M. Any subset of these column is either linearly independent or linearly dependent; the subset thus fall into two classes. These classes are not arbitrary; for instance the two following theorems must hold: a) any subset of an independent set is independent; b) if N_p and N_{p+1} are independent sets of p and p+1 columns respectively then N_p together with some column of N_{p+1} forms an independent set of p+1 columns […] Let us call a system obeying a) and b) a "matroid". (p. 509)(Whitney 1935). Hence the description with matroids will be very general (and quite poor – as Whitney: "The fundamental question of completely characterizing systems which represent matrices is left unsolved") but it has the great advantage of onlycharacterizing the relationship between elements in two modes: independence and dependence. And this remains valid for a large set of objects – Whitney: "In place of a matrix we may equally well consider points or vectors in a Euclidean space, or polynomials, etc…" and more recently: graphs, matrices, groups, algebraic extensions… (for a pedagogical introduction to matroid, see(Neel and Neudauer 2009)).

In matroid, *independence is not based on a specific type of relation.*We don't need to specify the relation: it is enough to know that there is such a relation as soon as, for a finite set E, there is a collection I of subsets of E which can be called the independent set of a matroid on E, M(E) if and only if I satisfies the following properties: i) I is non-empty; ii) every subset of every member of I is also in I (I is hereditary) ; iii) if X and Y are in I and $|X|=|Y|+1$ (the operator $|\ldots|$ designates the number of elements in a set of elements), then there is an element x in X-Y such that $Y \cup \{x\}$ is in I (independence augmentation condition).

Hence this paper aims at understanding the '*endogenous*' dynamics of technologies by applying design theory on a knowledge structure modelled with *matroid* - in particular we expect to characterize in this model how design strategies depend on (and also change) dependences and independences in a knowledge structure and how specific design strategies increase generativity and genericity.

3 A C-K MODEL WITH MATROIDS

3.1 Notions of techniques, systems and independences in a matroid framework

To build a C-K model with matroid, we introduce the matroid of known techniques as the K-space in C-K. Assume E, a list of known technique (also called technological building blocks in the paper): T_1,… T_n. We select a subset of this set of technological building blocks. If they build a working system[1], we will say they are *dependent* (we could say "compatible"); if not, they are *independent*.
In general, a set of technological building blocks is*not* necessarily creating a working technical system – in matroid terms: the elements are not necessarily dependent. Still we could consider one particular case, a "lego-like" matroid of technological system, in which every set of two (or more) building blocks is said dependent (ie it creates a working "lego") – this corresponds to a specific matroid, called the uniform matroid $U(1, n)$ (where n is the number of elements in E). Hence a first result*: a simple "lego-like" combinatorial model corresponds to the uniform matroid*.
To give a "visual" example: suppose that the matroid of techniques is graphic. It means that it can be represented by a graph G where each technological building block is an edge t_i (E = E(G), ie the set of edges of the graph) and the cycles of the matroid are the cycles of the graph. Let's add one additional element (that is not necessary in the matroid approach but is illustrating how the "technological approach" can be linked to a "customer-driven" one): consider the vertices of the graph, V(G); each vertex is a certain functionf_i; an edge t_i represents a technique (a technological building block) to address a pair of functions. Following the definition above, a *workingsystem is a circuitin the graph* (a path of technological building blocks that is connected and all vertices are of degree 2, ie the circuit goes only once through each vertex) – this workingsystem addresses all the vertices in the path (ie all the functions). This means a set of building blocks will be considered as a "working technical system" if and only if there is a *circuit*that links them all.
It is important to underline that this model fits perfectly with our requirements on a model of techniques (see 2.1.): in a matroid model, a technique, ie an edge, is both characterized by the vertex it links (eg. its functional role) and by its relations with the other techniques – ie whether it builds dependent or independent sets with other edges, ie whether it can be included in a working system with other techniques.
For instance the graph G below can be interpreted as a synthesis of the technological know-how of adesigner. The designer knows how to address $\{f_1; f_2; f_3\}$ (with the circuit t_{12}-t_{23}-t_{31}); he doesn't know any solution to address $\{f_3; f_5\}$; he doesn't know either how to address $\{f_4; f_5\}$: there is a path joining f_4 and f_5 but there is no circuit. A matroid can be associated to this graph of designer's knowledge, the matroid defined by the cycles of the graphs. In this matroid $\{t_{12}; t_{13}\}$ is independent whereas $\{t_{12}, t_{13}, t_{23}\}$ is dependent. Note that $\{t_{12}, t_{45}\}$ is also independent.

[1] We have here a terminology issue. We will distinguish two "systems", inherited by two disciplines. On the one hand, Bertrand Gilles, as an historian, speaks of "technical system" to designate the set of all techniques and their relationships. This is what we model with a matroid, the matroid of all known techniques. On the other hand, the authors of engineering systems speak of a "system" to designate a working assembly of techniques – this is a subset of the set of techniques, this set having the "working" property. This is what we call a "working system" in the matroid model. This is actually a circuit of the matroid. The "smallest" circuits are the cycles (see table 1)

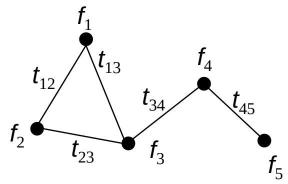
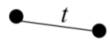
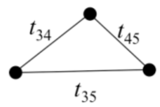
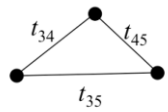
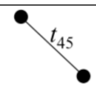
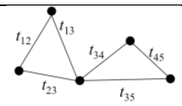
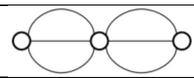


Figure 2: A graph G

We model in matroid basic notions of the dynamics of technologies: a technique, a working system, a family of techniques, the structure of all techniques, the structure of all working systems (See below).

Table 1. notions in the dynamics of techniques and notions in matroid theory

Dynamics of techniques	**Matroid theory**	**Illustration with graphic matroid**
Technique	An element in a matroid	
Working system: a system made of compatible techniques (techniques that work together)	A circuit (eg a cycle, ie a minimal circuit)	 extracted from M below
A family of techniques: a subset of techniques such that no "external" technique is compatible with the techniques in the family	A flat, extracted from a matroid	 or:  extracted from M below
The structure of techniques	The matroid of techniques	Matroid M: 
The structure of working systems	The dual of the matroid of techniques	

Note that matroid theory provides immediately one quantifier: a matroid has a certain rank, which actually corresponds to the size of the largest independent set. In a graph *G,* we have the rank function r(M(G)) = |V(*G*)|-1. (r(*G*)=4 in the example below). The rank corresponds to the highest number of independent building blocks: at most the graph *G* below enables to gather (in a set) four independent edges, hence four independent technological building blocks.

3.2 Designing a new matroid: a C-K design process on matroids

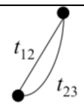
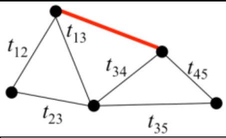
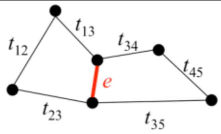
How does C-K apply to a knowledge base made of a matroid? We have to see how each classical notions of C-K are applied in case of a matroidal knowledge base.

In C-K theory, K is the space of propositions that have a *logical* status. *The logic in matroids is based on dependency*: the proposition "there is a working system including technique t_i" is true if and only if there is a circuit *C*containing t_i. For instance t_{12} is dependent of t_{13} and t_{23}, hence the proposition "there is a circuit that contains t_{12} in the matroid M(G)" is true; the proposition "there is a circuit that contains t_{34} in the matroid M(G)" is false.

In C-K theory, C is the space of propositions that is *interpretable* in K and *undecidable* in K. Interpretable: if K is modeled as a matroid, the proposition must be expressed in matroid language, hence it is a proposition on some of the building blocks of the matroid. For instance: "there is a technique (ie an edge) such that there is a circuit that contains t_{34}". The latter proposition is interpretable. It is also undecidable – it is neither true nor false in the matroid. One design for this new edge might be for instance an edge linking f_3 and f_5 (note that it is not the only solution: designing t_{25}, linking f_2 and $f_5$would also realize the concept). Hence we can formulate a concept for a matroidal K-base in C-K.

What is now the design process? Matroid theory teaches us that there are only*three basic operations* – design by deletion-contraction, design by extension and design by co-extension(Oxley 2011) – see table below. We will show (Le Masson et al. 2015) that these operations in matroid correspond to three ways to design techniques: designing one working system, designing one new working system compatible with the other techniques, or designing a generic technique (see synthesis table below).

Table 2. Design operations in the dynamics of technique and in matroid

Designing one working system based on existing techniques	Extracting one circuit (possible for all minors of the matroid of techniques)	
Cumulative design of working systems with new technique linking other techniques and minimizing propagations	Extension ie one dependent edge, depending on the techniques to be linked together	
Designing a generic technique, generic to several technical families	Coextension, ie one independent edge common to several connected components	

3.2.1 Trivial cases: deletion and contraction

To begin, let's first analyze the classical design issue that consists in designing one "working system" by using existing technological building blocks, while respecting their interdependences. In matroid terms, it consists in "extracting" one specific dependent set from the initial matroid by relying on two operations that keep the matroid structure, namely deletion and contraction. Actually the "systemic impact" of this process is not limited to the extraction of one working system (ie a cycle in the matroid) but more generally it enables *to extract some subsets of technological building blocks from the initial set of technological building blocks* (ie a sub-matroid extracted from the initial matroid – this subset is called a "minor": a minor is a sub-matroid that can be deduced from the "big" one by deletions and contractions.)

How can one describe the design process? Deletion and contraction in matroid generalize deletion and contraction in a graph – let's analyze the graph case:

- Deletion in a graph consists in skipping a edge. *Deletion is equivalent to decide to* not *use a technological building block for a certain design.* To *not* use this building block also means to not use certain cycle that use this building block. In the example below: to design the working system $\{t_{12}; t_{23}; t_{13}\}$ one deletes t_{34} and t_{45}.
- Contraction in a graph G occurs relative to a particular edge, e. The edge e is removed and its two incident vertices, u and v, are merged into a new vertex w, where the edges incident to w each correspond to an edge incident to either u or v. One writes G' the new graph and one writes G'=G/e. Contraction is equivalent, in linear algebra, to a projection in a space orthogonal to the contracted element. Contracting an edge in a cycle actually means that *a working system can be "reduced" to smaller one with less technological building blocks*(and less functions) (e.g. a refrigerator without cooling system becomes an isothermal box). For instance the concept "a circuit that contains only $\{t_{12}; t_{23}\}$" is obtained by contracting t_{13}(and deleting t_{34} and t_{45})

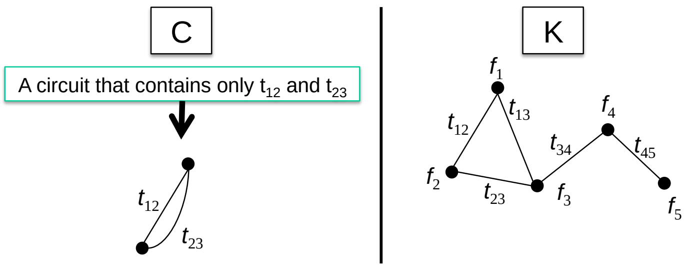


Figure 3: A design by deletion and contraction

One very famous theorem in matroid theory, the "scum theorem"(Crapo and Rota 1970), shows that the formation of a minor N from a matroid M can always be obtained in asimple two-stage process: a contraction of M (to get the rank right) followed by a deletion to remove the remaining elements not in N. Hence a very simple design process to "extract" one working system from a matroid of techniques.

More interesting are cases where the concept actually tends to design a new "bigger" knowledge structure – ie the concept addresses the *whole* matroid, and not only a subpart of it and the process will “grow” the initial matroid M, in such a way that the new matroid N will “contain” the old one. For these non trivial cases, matroid theory tell us *two ways to design new edges in a matroid: extension and coextension*.

3.2.2 Extension – or the design of a non-pervasive technique

The most intuitive is called an extension. It is the reverse operation of a deletion: if a matroid M is obtained from a matroid N by deleting a non empty subset of E(N) then N is called an extension of M. In particular if the subset is a simple edge {e} then N is a called *single extension* of M (see the example in the figure below, we add t_{35} as a new edge in M(G)).

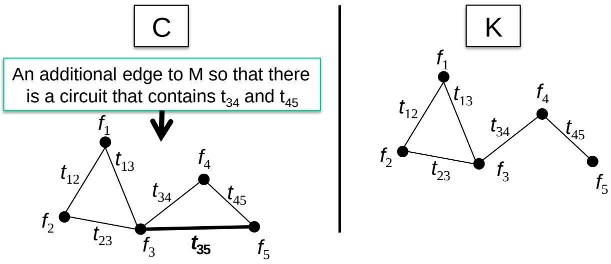


Figure 4: A design by extension

1- What is the systemic impact of extension?

To analyse the systemic impact, we need one additional notion from matroid theory, namely the notion of *flat*: a flat is a set of elements of E such that it is impossible to add a new element of E into it without changing its ranks. In the matroid Min figure 1, {t_{12}, t_{23}, t_{13}} is a flat and {t_{12}, t_{13}} is not a flat; {t_{34}, t_{45}} is also a flat. One says that a flat is a “*closed*” set in a matroid. A flat can be seen as a “family” of building blocks that are incompatible with any other building block outside the family – for instance we can consider that the family of techniques used in automotive industry is a flat because these techniques are said incompatible with any other techniques from another industry. The set of flats forms a lattice (with the inclusion relation). This lattice of flat represents the families of technologies and their inclusion relations. For instance one represents below (figure 4-a) a matroid and its lattice of flats.

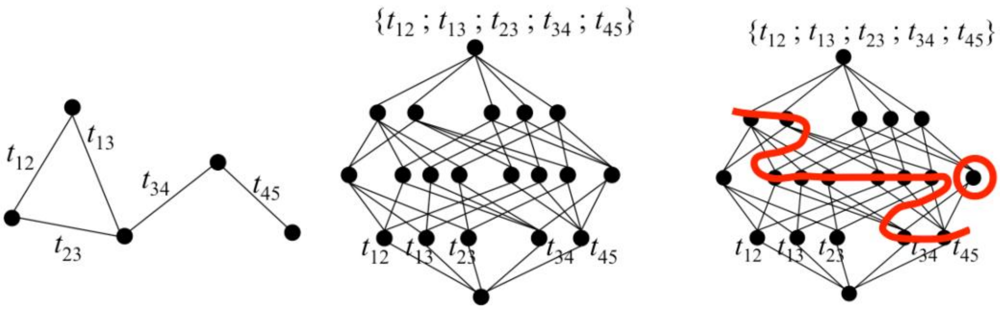


Figure 5: a) a matroid b) its lattice of flats and c) the effect of the extension t_{35} *on these flats*

The systemic impact can be modeled as the impact of the design of a new edge on the structure of flats. When M is extended with a new edge e to form the matroid N, we know from matroid theory that there are three possibilities for any flat *F* :

- *Type 1 “e-independent flats”:F* U e is a flat of N and r(*F* U e)=r(F)+1 – it means that *the new edge is independent of the flat*. If t_{35} is added in G above, then {t_{12}, t_{23}, t_{13}} is such a flat in G. These are all the flats that are *not “impacted”*by the design of e. More generally these are the flats below the red line on figure 4-c.
- *Type 2 “e-determining flats”*: *F* U e is a flat and r(*F* U e)=r(F) – it means that *the new edge e is actually dependent on the edges of F*. If t_{35} is added in G above, then {t_{34}, t_{45}} is such a flat in G. These are the flats above the red line on figure 4-b. Note that these flats follow a certain structure: they form a so-called modular cut, which means that they are ordered by inclusion

and *there is a smaller flat, which is in included in all other type 2-flats*(in our case:{t_{34}, t_{35}}) (the demonstration would require more space, see theorem 7.2.3 in (Oxley 2011)).

- *Type 3 "e-determined flats":F* U e is not a flat of N – it means that some other elements should be added to *F* U e to "close" it into a flat, i.e. there are "old" edges of M that are now dependent on elements of *F* U e. *e has created new dependancies*. One example if {t_{34}} in G: {t_{34}, t_{35}} is not a flat, since t_{45} can be added without changing the rank. These are the flats on the red line in figure 4-b.

Hence the impact of extension on the lattice of flat: 1) the extension "keeps" several flats unchanged (type 1 flats) . 2) The effect is confined to one "smaller" flat (here {t_{34}, t_{35}}) and all the flats that contain it, and in these flats the extension does not change the rank – hence it creates new dependences (same rank with more edges), ie it creates new cycles, ie it creates new "working systems". *Extension corresponds to the design of a technique that creates new systems inside one family (and inside all the families that contain it) and is non-pervasive for all the others. This is a non-pervasive technique that creates new working systems.*

Let's remark one special case: the "minimal" non pervasive case occurs when there is only one type 2-flat; hence, necessarily, this single type-2 flat is E itself (all the edges). It means that e is creating a cycle with the biggest independent set (without any smaller cycle inside). This is called a "free extension" – no other flat than E is changed in this extension.

2- What is the design strategy? The concept: designing a new working system.

With the elements given above, it appears that extension actually corresponds to *one* new dependenceinside one target family of techniques (the smaller type-2 flat). Hence an extension corresponds to a *concept:* "given (at least) two independent techniques in the matroid of techniques M, designing an edge e that is added to M to create in M a new working system that contain these techniques".

This very simple result actually also helps to understand very important distinctions in the notion of "combination" when dealing with the dynamics of techniques:

1) *we can distinguish between a "new" working system (a "new combination") and a working system that was deducible from the existing techniques (a "known combination"):* extension corresponds to the design of one *new* working system; the new working system was *not* in the matroid of techniques, *it was not "decidable" in the initial matroid*. Otherwise the working system was not a concept, it was already true in the matroid.

2) *we can distinguish between the design of a new working system (by deletion-contraction) (combination without taking into account the interdependencies with other techniques)and the design of a new set of techniques that contain a new working system, ie the design of a new working system taking into account all the interdependencies with previously known techniques (by extension and co-extension) (combination that takes into account the interdependencies with other techniques)*. In our case, the former case would correspond to the design of {t_{34}, t_{35}, t_{45}}, "extracted" from M ("forgetting" all the properties of dependence or independence with all the techniques); by contrast an extensionis an extension of the matroid of techniques itself. An extension is driven by the design of one new working system but it also "controls" the impact on the whole matroid of techniques. The final result of the design is *not* limited to {t_{34}, t_{35}, t_{45}} but is the matroid MUe (that contains in particular the working system {t_{34}, t_{35}, t_{45}} but also all the dependences and independences).

To conclude on extension, its design and its systemic impact: *extension models the impact of the design of one new working system on the set of techniques – and more precisely it models the design of a "non-pervasive" technique*.Extension models the cumulative, non-pervasive creation of new working systems.

We could say that a good engineering department should design by extension: based on (the matroid of) known techniques, it designs one new technique to get a new working system, it cumulates the knowledge techniques and designsthe new technique for the new working system by minimizing the impact on the other techniques.

Note that we better understand how engineering department deals with combination:

a) It invents a new combination that did not exist before (there is a new edge) – this is different from the identification of a combination that was already known.
b) This combination takes into account the interdependencies with all other techniques – this is

different from an opportunistic extraction of techniques without taking care of compatibilities.

3.2.3 Co-extension – or the design of a technique with systemic impact

Co-extension is a less intuitive edge creation process in a matroid. This is the reverse operation of a contraction (see above). In matroids, N is a coextension of M if there is some set T such that M=N/T. Interestingly enough a contraction operation in matroid is analogous to a *projection* (along T) (see Oxley (Oxley 2011), chapter 3.3). *Hence coextension is analogous to the reverse of a projection* (see Oxley (Oxley 2011), chapter 7.3), ie an *expansion*.
This operation can also be seen as an *extension of the dual* of the matroid: if N* is an extension of M*, then N is a coextension of M. And we know quite well now what extension means in a matroid. Hence understanding co-extension requires to understand what is the dual of a matroid. What is the dual M* of a matroid M? Formally speaking, M* is defined over the same set of edges as M; it can then be defined by its bases: given a basis B of the set of basis of M, a basis of M* is the set obtained by E(M)-B (this definition is acceptable because it can be proven that the set of all the "complementaries" of basis of M forms a set of basis of a matroid, that is called M*). In the case of a graphic matroid M(G), the circuits of M* are formed by the sets of edges that "connect" the graph, ie each of these sets corresponds to edges that, once removed, separate a new connected component in the graph. The dual of M can also be defined by *the connected components* (or minimal cycles) of M: the dual of M is formed by all the edges linking connected components. In the particular case where the graph G is planar (no edges are "superimposed" in the plan), it is possible to represent the dual as in the figure below.If one studies a matroid of techniques,then *connected components are "working systems" (minimal one) hence the dual is the set of relations between (minimal) working systems.*

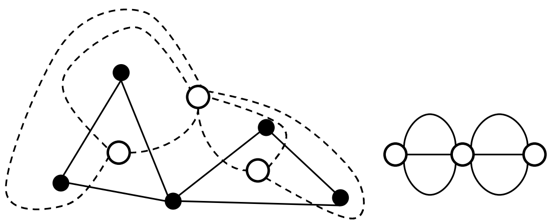


Figure 6: A graph and its dual. On the left, black dots: the initial graph G; in white dots and dotted lines on the left: the dual G of G. On the right, the same dual G* without the underlying G.*

Designing an additional edge in a matroid by coextension can be represented by the figure below: in K, the matroid and its dual; in C, three possibilities to extend the dual and to deduce the coextended matroid N. There are multiple possible extensions of a dual.

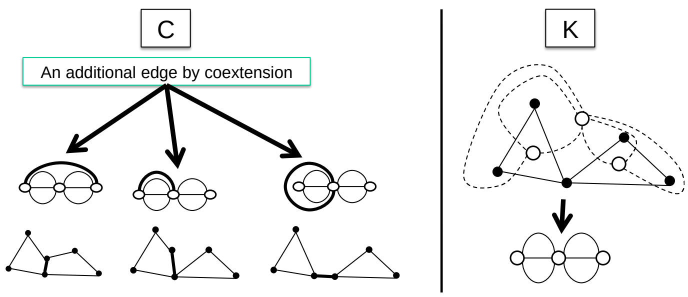


Figure 7: Coextensions of a graph G. On the right, black dots: the initial graph G, M=M(G); on the left three coextensions (and related extensions in the dual) –the left coextensions corresponds to the concept: "an additional edge so that there is a circuit containing all the edges (technological building blocks) of M"

We can illustrate the coextension logic, and its difference with extension, on the pratical example below. Suppose one knows techniques to make knives and techniques to make bottle openers. By extension one can take one technique of the "family" of knives and another from the family of bottle opener and make them dependent by designing one additional technique – namely the insertion of a bottle opener in the knife handle: this is an extension. A co-extension is a technique to keep all the previous knifes and bottle opener technique: the articulation of the tools on one handle is the technique that enables to design swiss army knives.

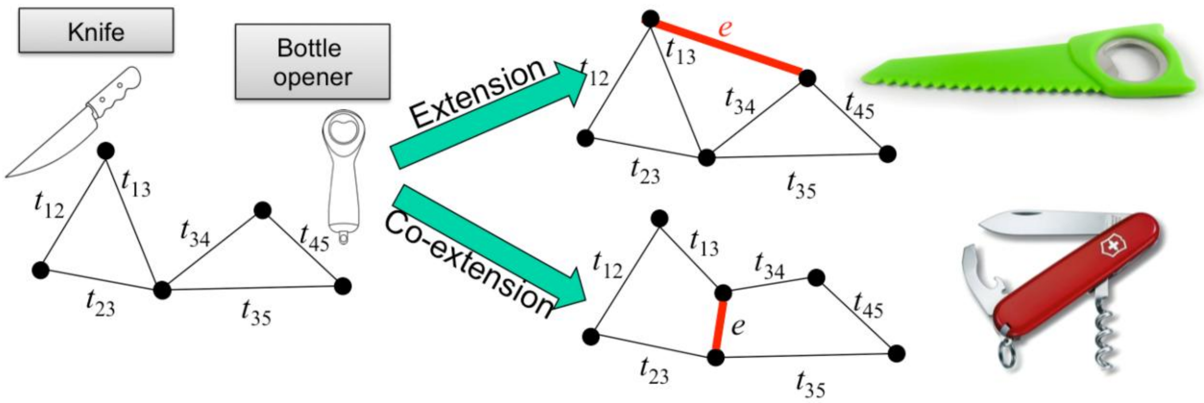


Figure 8: Illustration of the extension and co-extension on one simple case

1. What is the systemic impact of co-extension?

Let's analyse the impact of coextension on the matroid, contrasting it with extension (we won't demonstrate the properties below, they are relatively classics in matroid theory (see (Oxley 2011) – we rather insist on their consequences in the perspective of the dynamics of technologies):

c) Extension *preserves* the rank, creates a *new dependent* edge in a flat – hence extension creates a new working system using the new technique. We can consider that the new working system is a "*direct value*" of the extension. By contrast co-extension creates a *new independent* edge and *increases the rank*. This means that the new building blocks (the new edge) is not included in a "new" working system combining building blocks that couldn't be combined before. If one considers that the direct value of design is in the new working systems that use the new technology – then *coextension doesn't create direct value*! The value created by the new edge is not self-evident.

d) Extension is non-pervasive: it modifies only the flats that include the newly created working system. By contrast, coextension doesn't create a new working system with the new technique and, even worse, it disturbs "old" working systems! In the figure below we see that the coextension transforms a working system $\{t_{12}, t_{13}, t_{23}\}$ into an independent set (hence no more a working system) and it is necessary to add *e* to the system to make it work again. *The new technique is now required to make work systems that worked without it before! We could speak of value destruction.*

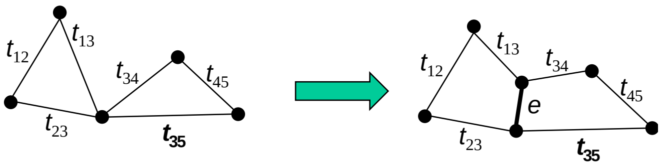


Figure 9: Coextensions of a graph G.

e) The new edge connects connected components that were independent before, ie the new technique enables "bigger" working systems by aggregating smaller working systems. The new edge integrates known working systems into a bigger one. Strangely enough the new, "bigger" working system does *not* contain the new edge! It enables to combine into one working system two previously known, but independent, working systems. This is the critical property of coextension: *it "combines" working systems*. As such it is *pervasive*.

f) Working systems combination is "*modular*". For one "old" working system (say $\{t_{12}, t_{13}, t_{23}\}$), there are now two working systems alternatives: $\{t_{12}, t_{13}, t_{23}, e\}$ and $\{t_{12}, t_{13}, t_{34}, t_{45}, t_{53}, t_{32}\}$. It means that $\{e\}$ and $\{t_{34}, t_{45}, t_{53}, t_{32}\}$ are interchangeable from the point of view of $\{t_{12}, t_{13}, t_{23}\}$. *Hence coextension creates what engineering usually calls "platform" and "modularity".*

2. What is the design strategy in coextension? The concept: designing a generic technique.

With the elements given above it appears that coextension consists in creating a relationship between (at least) two working systems to create a new working system that keeps all the properties of the aggregated systems (all previously known building blocks are present) and is modular. This is the concept behind coextension: *designing a technique that enables a working system that combines working systems – this technique will be common to working systems that were independent until now*. *Hence this concept corresponds to the design of a generic technique, a technique designed to be generic to several, ex ante independent, working systems.*

Let's underline two surprising properties of the generic technique:

a) As expected from a "generic technique", the new technique is *pervasive*, it has a *strong systemic impact,* it combines working systems and enables "bigger" systems. But paradoxically it is *not* visible in the newworking system! (eg, in the figure above: $\{t_{12}, t_{13}, t_{34}, t_{45}, t_{53}, t_{32}\}$). The new technique is "hidden" by the new modules and platforms. It only appears when one shifts from one module to another. Still we will confirm this key feature of generic technique on some cases: in steam engine history as well as in semiconductor techniques history, a generic technique is not a "big", new, breakthrough working system emerging out of nowhere – it is a discrete technique that helps combine existing working systems while changing them. It corresponds to a form of "creative destruction".

b) Based on results from matroid theory, it is possible *to increase the number of independent working systems combined by one new technique*. Hence the *genericity can be designed*. Genericity is not necessarily the result of the random aggregation of working systems. Based on proposition 7.3.9 in (Oxley 2011) it can be shown that a generic technique that would be generic to all "working systems" (ie dependent flats in matroid terms) corresponds to the design of a free extension in the dual of the matroid of techniques. We show below one such generic technique.

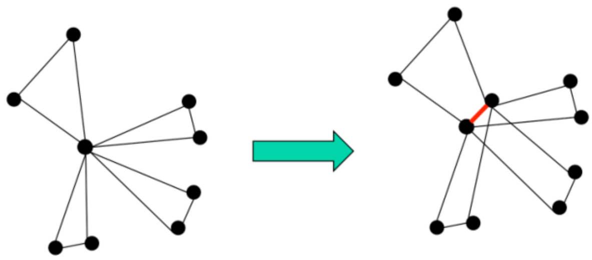


Figure 10: Designing a technique that is generic to four independent working systems.

To conclude this part: we built a C-K model with matroid to study the endogenous dynamics of techniques. *This model enabled us to considerably enrich the representation of this dynamics and the models of "combinations" in design:*

1-We show that the usual "lego-like" model of "technology combinations" actually corresponds to one very specific matroid, U(1,n), in which each technique is compatible with any other.

2-the model accounts for some critical distinctions in the dynamics of techniques and technical systems.

a) it accounts for the *distinction between deduction and construction*. Deduction consists in proving that one working system (circuit) exists in the matroid; construction consists in creating techniques to design one new circuit (a new working system).

b) it accounts for the distinction between designing a "stand alone" working system based on existing technique (one "project") and designing a structured set of techniques: the first one consists in extracting one working system from the matroid of known techniques to create an "independent" working system – and we can show that it is possible for all the "minors" of the matroid; the latter consists in enriching the matroid of techniques.

These distinctions enlighten different forms of combinations, from non-generative combinatorics to generative combinatorics:

a. The identification of one circuit that already exists in a matroid corresponds to combinatorics

b. Designing one artifact from existing techniques, by deleting some interdependencies with other techniques corresponds already to a (limited) generative combinatorics (some circuits are created that where not circuits in the initial matroid of techniques – but no new edges are created).
c. Extension consists in creating a new edge (new technique) to create a new working system, taking into account all interdependencies. This is a generative combinatorics: a new edge and a new working system are generated in the process. The new edge is actually dependent on the previously known one, its design is driven by a concept of working system.
d. Co-extension consists in creating a new edge that increases the dimension of the whole set of techniques, ie open new extension possibilities, ie create new opportunities to invent new combinations. It also a generative combinatorics.

3- We show that the two fundamental processes to "enrich" a matroid with a new edge correspond to two different design reasoning:
- *extension consists in designing one new working system in a non-pervasive way* (minimizing the systemic impact). It can be compared to the activity of an engineering department regularly designing new working systems in a cumulative way, avoiding the propagation of changes to all its technological building blocks.
- *coextension consists in designing a generic technique enabling the (pervasive) combination of previously independent working systems*. It can aim at combining the maximum number of working systems, hence it aims at maximizing its genericity. It doesn't create "direct" value (the new technique is not involved in any new working system!), it is disturbing the "old" working system, it recombines the working systems in a modular way. It corresponds to the design of generic technique – an activity that is very difficult to analyse but is becoming more and more critical at the age of platform competitions and pervasive techniques.

3.3 Historical illustration: the genericity of steam engine modelled with matroids.

With the illustration below we show how matroid models help to account for one famous historical case of designing a generic technology and how it confirms the most paradoxical properties of generic techniques modeled in matroids.
The story of the steam engine is often told this way: Watt designed a steam engine and progressively many applications were found for it. Yet, as shown in (Dickinson 1936; Thurston 1878) and analysed in detail in (Kokshagina et al. 2013a; Kokshagina et al. 2012), this story does *not* correspond to how Watt and Boulton designed a "generic" steam engine.
Actually the first generation of steam engines was adapted to mining, but not to other uses; hence there was no "steam engine" in the 1770s, but only *water pumps for mining* – and Watt was first famous, in 1770s for greatly enhancing Newcomen *"fire pumps"* with a separate condensation chamber. Actually the story of the "*generic* steam engine" begins later: in the 1780s, Boulton asked Watt to work on a new concept "a steam engine that is compatible with multiple machine tools" and it is Watt's new design for this concept, that appeared as the "steam-engine generic technology". Boulton's brief was targeting a complete recomposition of the structure of techniques of his time: if a steam engine is compatible with multiple machine tools, it becomes a core component of all future machines, as all new machine tools will be redesigned to take the best advantage of the steam engine. This "genericity" goal corresponded for Watt to a surprisingly focused design issue: design a new *generic*technique to transmit movement from the steam engine to any other machine. This is what led Watt to the well known the "reciprocating steam engine" (see figure below).

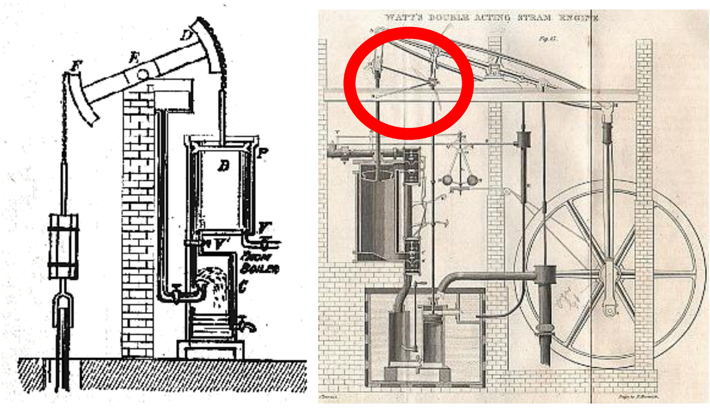


Figure 11: a) 1763 Watt steam engine with separate condensation chamber (not generic) ;
b) 1784 Watt & Boulton Double acting steam engine. The parallelogram creates genericity.

In C-K/Ma the design of a "generic" steam engine relies on co-extension. In K: the known technique. In C, Boulton concept, and the design result: a new order of techniques based on a new technique for movement transmission. The latter appears as a "*platform*" that connects the steam engine either to mining or to workshop machine tools (in textile, iron industry, etc.). The main design effort consisted in *designing the coupling technique* (here the double acting transmission system). An extension logic would have led to adapt the steam engine to each of the applications. The co-extension consists in designing one new technique that enable to combine steam engine with many applications at once.

This example illustrates many features (and paradoxes) of the design of generic techniques:

- Generic technique doesn't emerge as a complete original technology – it actually relies on existing techniques.
- Despite their great systemic impact, they are discrete: the invention is neither in the fire engine, nore in the condensation chamber, it is only in the cinetic mechanism on top of the vertical rod
- This is not an evolutionary process in which designers "discover" phenomena or "combine" randomly techniques: this is the intentional design of a pervasive technique(Hooge et al. 2014).

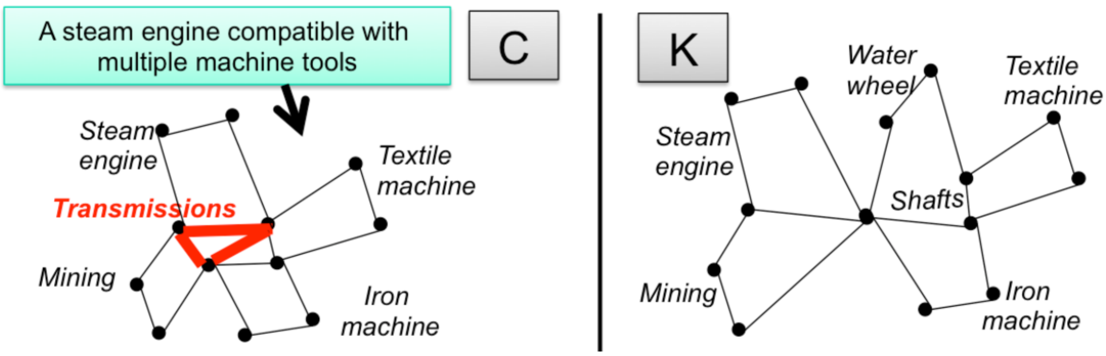


Figure 12: Designing steam engine as a generic technology

4 A COMPUTATIONAL MODEL OF THE DYNAMICS OF TECHNOLOGIES

We show now how C-K/Ma provides us with quantifiers for critical features of the dynamics of technologies (dependence, genericity, generativity), helps to analyse specific cases and finally characterize specific dynamics.

4.1 Characterizing features of the dynamics of technologies

The matroid approach helps to quantify and characterize complex notions like independence level, generativity and genericity.

4.1.1 Independence

The literature review helped to realize that the endogenous dynamics of technologies depends on the independence between techniques. Dependent techniques are already linked by strong relations, they can be deduced one from the other – they are deterministic. Note that in a Lego-like model, all combinations are considered as known. On the other hand independent techniques are *not yet* related so they are “holes” in the knowledge base.
With C-K/Ma we can first *quantify independence and dependence*: the rank of the matroid in K gives the level of independence; the co-rank gives the level of dependence (The level of dependence is also the independence in the dual, which actually corresponds to the independence between working systems.). Both are linked by the equation: $r(M) + r(M^*) = |M|$ where $|M|$ is the number of edges, ie the number of technological building blocks.
Let’s underline two special cases:

- “pure combination”: In a Lego-like matroid (U(1,n)), the rank is one – all techniques are dependent on the others; it means that there is no “holes”: no two independent techniques; the dual is U(n-1,n), ie n-1 working systems are necessary to “deduce” the last n^{th} one. Ie the systems are independentfrom each other.
- “perfect engineering system”: Let’s analyse another “intuitive” model of systems. We consider now the n techniques in a working system that follows the first axiom of Suh. This is a working system, hence the n techniques form a circuit. Now the first axiom applies “inside the working system”, which means that the dependence linking all techniques to form a working system is neglected and any subset of n-1 techniques follow the first axiom. Hence the set of techniques enabling a Suh working system might be represented by a U(n-1, n) matroid.

Now we also want to quantify the effect of independence level on generativity and genericity.

4.1.2 Generativity

Generativity is a critical feature of any design theory (Hatchuel et al. 2011). It is usually hardly quantified. In C-K/Ma it is possible to give a quantified evaluation of the generativity associated to a matroid Mand it depends on M independence level (ie M rank). Generativity can be seen as a quantification of the number of new edges (techniques) that can be created on a given matroid. We know now that they are two fundamental processes of single edge creation –hence we can *quantify the generativity by the number of possibilities of single edge extension $g_{ext}(M)$ and the number of possibility of single edge coextension $g_{co\text{-}ext}(M)$ (see figure below):*

a. Generativity by (single edge) extension: it is possible to evaluate the number of single edge extensions that can be done on a given matroid of rank *r* ($r(M(G))= |V(G)|-1$) and for $n=|V(G)|$ vertices. The maximal number of edges is n.(n-1)/2 (this is a so-called *complete* matroid, where every vertex is linked by a single edge to any other vertex). If the matroid is simple (no loop, no parallel edges) there are already $|M| = r+r^*$ edges in the matroid (where r* is the corank, ie the rank of the dual of M), hence **the number of possible extensions for a simple matroid M is $g_{ext}(M) = r.(r+1)/2-|M| = r(r-1)/2-r^*$**. If the matroid is not simple, then there are p loops or parallel edges hence the simple matroid associated to M has $|M|$-p edges and the number of possible extensions for a matroid M with p loops and parallel edges becomes **$g_{ext}(M) = r.(r+1)/2-(|M|-p) = r(r-1)/2-r^*+p$**.This is one first measure of the number of “extensions” (ie cumulative, non-pervasive design of a new working system) with a given set of techniques. *This could be assimilated to the growth potential of an engineering department knowing the matroid of techniques M.*
b. Generativity by (single-edge) co-extension: on the other hand it is also possible to evaluate the maximal number of coextensions that can be made from a given graph of rank r, with $|M|$ edges. We have **$g_{co\text{-}ext}(M)= r^*.(r^*-1)/2-r+p^*$** (where p* is the number of loops or parallel edges in the dual of M). *We estimate here the quantity of “generic” techniques that can be proposed on a set of given techniques*. Note that this is usually far from negligible.

4.1.3 Genericity

With genericity we mean the number of applications *derived from one design*(Kokshagina et al. 2013a). *Genericity characterizes the systemic impact of one particular new technique*. In case of matroid, genericity can be modeled as the set of new circuits resulting from the design of a new edge *e*. We can characterize two contrasted forms of genericity (see figure below):

a. We count the *new circuits created* by the design of the new edge e. In extension, the new edge creates many new circuits, in the target flat and in all the flats containing it. In coextension, we need also to count all the circuits created by the design of e and not involving e directly. This genericity results from the new combinations between connected components of technological building blocks. Since a circuit is a working system and a working system can be considered as marketable, *this genericity can be assimilated to the "direct value" created by the new technique*.

b. We can also count the new circuits that can *now* be created (after the design of *e*), with an additional effort - an extension effort or a coextension effort. In this case this is *the "indirect value" or the "dynamic value"*. We call M the initial matroid and N the matroid created by e added to M. We will use the generativity quantification constructed above. Note that in extension and in co-extension the number of loops and parallel edges is unchanged, ie p(N)=p(M) and p*(N)=p*(M). For the edge e we compute extension-genericity: $gen_{ext}(e) = g_{ext}(N)-g_{ext}(M)$ (and we can conversely compute: $gen_{co\text{-}ext}(e) = g_{co\text{-}ext}(N)-g_{co\text{-}ext}(M)$). We distinguish two cases:
- If *e* results from an extension, then we know that r(N) = r(M) and r*(N)= r*(M)+1. Hence the extension generativity has decreased by -1 with the design of e by extension. $\mathbf{gen_{ext}\ (e_{ext}) = -1}$. (Similarly $gen_{co\text{-}ext}\ (e_{co\text{-}ext}) = -1$).
- If *e* results from a co-extension, then we know that r(N) = r(M) + 1 and r*(N)=r*(M). Hence $gen_{ext}(e_{co\text{-}ext}) = (r+1)r/2\text{-}r^*\text{+}p - [\ r(r\text{-}1)/2\text{-}r^*\text{+}p\ ]$, ie $\mathbf{gen_{ext}(e_{co\text{-}ext}) = r}$ (conversely: $gen_{co\text{-}ext}(e_{ext}) = r^*$). *The extension-genericity of e created by coextension is approximately r*. conversely the co-extension genericity of e created by extension is approximately r*.

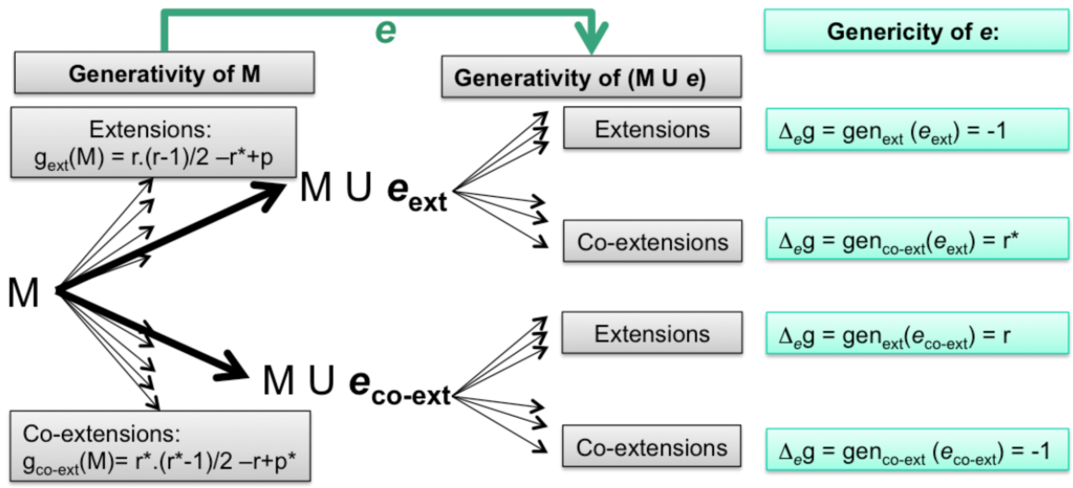


Figure 13: Generativity of a matroid, genericity of a new edge added to this matroid

4.2 Illustration: designing generic technologies in the semiconductor industry

We use these quantifiers to analyze the design of generic technique in semiconductor industry. It is presented in more detail in (Kokshagina et al. 2013b). It can be schematically described as follows: there are semiconductors systems that integrate radio signals (radar, wifi,…) and computing power; but these systems are poorly integrated and neither use high level radio frequency sensors nor powerful computing power. The concept consists in combining three, initially unrelated, technological families:

- Family 1: the computing technological building blocks (technological building blocks for so-called CMOS transistor)
- Family 2: the Radio-Frequence sensors, able to receive and digitalize radio frequency signals, these RF system being based on bipolar technologies

- Family 3: the so-called “back-end” system, the building blocks basically in charge of routing and processing signals

The concept is: “a matroid of technological building blocks that combine the three technological families”. By combination one expects a system that enables circuits using building blocks in the three technological sub-systems (computing, RF and back-end) and circuits using building blocks in every pair of the three technological sub-systems (RF and computing; RF and back-end, computing and back-end). The notion of “combination” is usually quite fuzzy and hides the design issues. Applying the matroid model enables to clarify several contrasted alternatives with different “values”. Combining extensions and co-extensions, one can propose four different solutions (see below) and evaluate their potential in term of *generativity* and hence *compare the genericity created by each design alternative*.

- The “pure extension” appears as an effort *to design “micro-combinations” of building blocks*; it brings multiple working system –hence a direct value-, still no one of these systems enable to combine *all* previously known building blocks. The genericity is negative: it means that the new design has decreased the generativity potential of the matroid.
- The “pure co-extension” is a generic technology that creates a working system that uses all previously known techniques; moreover it is modular (pairwise comparisons are also possible). Hence a direct value here also – but the number of newly created working systems in co-extension is rather lower than in “pure extension” case. The genericity is very high: it means that the new design has increased the generativity potential of the matroid.
- Hybrid case 1is also an interesting design strategy: it exhibits a new connected sub-component that connects each of the modular ones. We see here a “*platform*”. Genericity is relatively high.
- Hybrid case 2 enables to get the expected performance (connections between the connected components) but it does so *by designing only two edges*, whereas all the other solutions design three edges. This is a particularly efficient “generic” technology.

The design strategy adopted at STMicroelectronics to develop the new technique actually follows that process: mixing sub-technologies of CMOS and bipolar into a new connected component called bi-CMOS (hence an extension) and the redesign of the back-end to connect it to the new bi-CMOS.

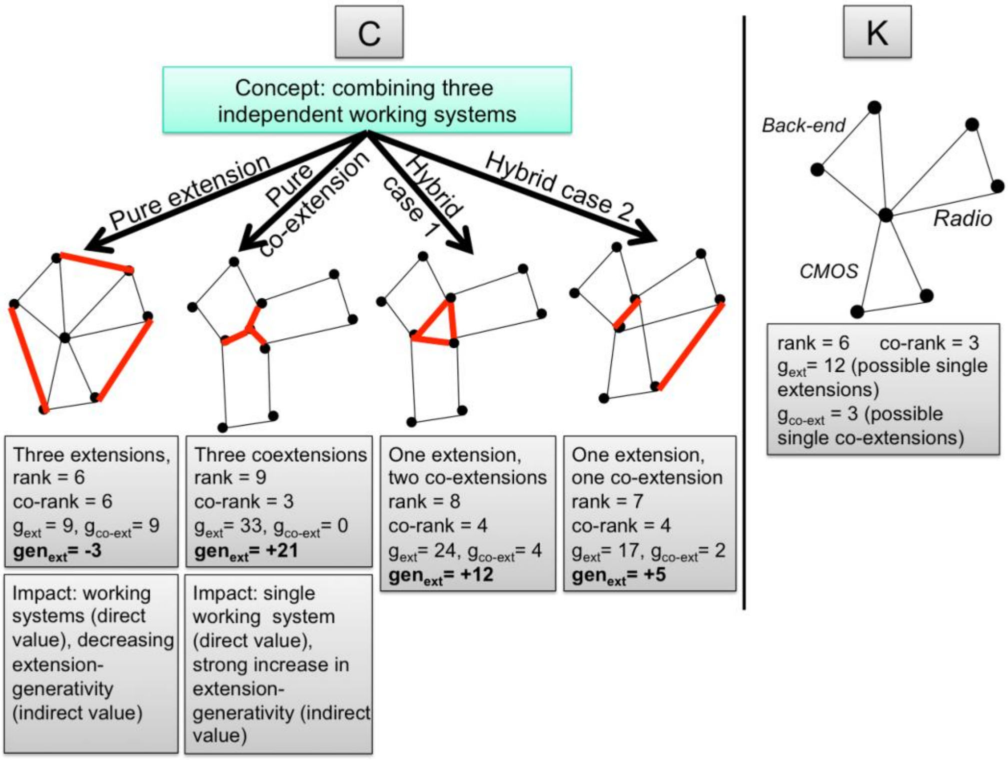


Figure 14: Combining technologies – the case of B9MW in semiconductors

This example shows how the C-K/Ma enriches our understanding of the design of techniques with systemic impact, supported also with quantitative criteria on generativity of the new matroids and genericity of the designed technique.

4.3 Characterizing the dynamics of techniques

C-K/Ma also uncovers general rules on the dynamics of techniques.

4.3.1 Combining basic processes (deletion-contraction, extension and co-extension)

We have studied two fundamental processes for adding an element to a matroid. Of course it is possible to combine the operations above (deletion, contraction, extension, co-extension) in more complex cases. We give an illustration below.

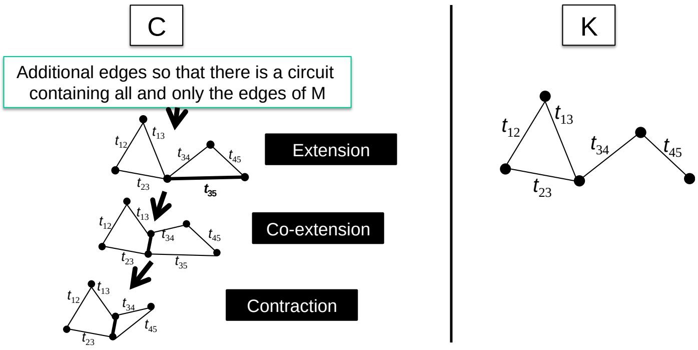


Figure 15: A design process combining basic matroid design operations

4.3.2 One general, negative result: Lego-like structure of techniques implies exogenous dynamics

Let's analyse the "intuitive" model of "combination" (used for instance in (Arthur 2009)). In this model, each building block can be combined with another one to create a working system – hence we called it the "legolike" structure. As seen above, this structure is the matroid U(1,n). We have (self-evidently): g_{ext}(U(1,n)) = 0 : extension is impossible in U(1,n). *This system is "locked" for extension*. The pure combination approach prevents the design of new working systems! (actually because all working systems obtained by combinations are considered as known in the lego-like structure – ie any working system, ie any circuit linking techniques from U(1,n) can be obtained by deletion-contraction).

Still we can compute: $g_{co\text{-}ext}$(U(1,n)) = (n-1).(n-2)/2-1. Hence U(1,n) is very generative in *coextension*. However, these coextension are hardly visible in "lego-like" approaches *because the co-extended matroid is no more Lego-like* (the U(1,n) family is not stable by coextension). *Hence a Lego-like model of the dynamics of technologies won't be able to account for technological dynamics based on coextension.*

Hence the extension of Legolike structure is locked and coextension can not be modeled. *Hence a Legolike structure can not account for the endogenous dynamics of technological systems.*That's why intuitive "combination" models tend to rely on exogenous dynamics.

This result is actually quite general: since coextension is not always intuitive, one tends to focus on extension – and extension, as seen above, has a negative genericity. So that there is a finite number of extensions. When this maximum is reached (all "simple combinations" are invented), and in the absence of coextension, the dynamics of techniques relies on the exogenous emergence of a new independent element in the matroid.

4.3.3 Another general, negative result: a structure of purely independent techniques implies exogenous dynamics

In the case of an "ideal" engineering system following Suh axiom, we showed above that this system might be represented by a U(n-1, n) matroid. Let's analyse genericity: it appears immediately that in such a matroid, no co-extension is possible. Ie *Suh's first axiom prevents the design of generic techniques*.

More generally, in system engineering one tends to favour systems with independent techniques – for instance, one will tend to separate mechanical, thermic, chemical or biological processes in a complex system (like a boat or a process machine). Interdependences are rather defects, eg. chemical corrosion of mechanical parts in a boat. *Hence a "perfect" system engineering will finally prevents the* *emergence of a generic technique*! Conversely by co-extension U(n-1; n) is not stable, so that the newly created structure is nomore following Suh Axiom. Hence here again it is not possible to account for endogenous dynamics if one considers that the techniques follow the Suh axiom of independence. Here again, the dynamics of technical systems is often modelled by the introduction of a new independent element in the matroid – ie an independent technique.

Hence the two simplified representations of technical systems – pure combination or pure independence- actually correspond to very particular matroids and they are actually representations that can not account for the dynamics of techniques.

4.3.4 A theorem for continuous endogenous dynamics: the necessary combination of extension and coextension

More generally, it also appears that *extensions or coextensions alone lead to deadlocked systems since the extension-genericity of extension is negative (decreasing generativity) and the coextension-genericity of co-extension as well*. We might have here an explanation for the “blocked technical systems” described by Bertrand Gilles (Gilles 1986).
A direct consequence of the negative genericity is that the only way to get an unlocked endogenous dynamics consists in combining extension and coextension – ie the combination of the design of working system and the design of generic techniques.

Let’s give some elements on this extension-co-extension dynamics:

- We can notice that extension-generativity decreasing by one after extension whereas,after a co-extension, extension-generativity increases by *r*. Hence we would expect much more extensions than co-extensions. Hence we identify two regimes:
- The “*extension-driven” regime* gives priority to extension (the design of working systems). In this regime, co-extensions (the design of generic techniques) are as rare as possible. Over time the matroid becomes saturated and no extension is possible anymore. Hence one co-extension is required, it increases the rank by +1 (the rank becomes r+1)and the generativity by +r. Over time the rank increases slowly (one co-extension that increases the generativity by r and the rank with +1, then r extensions until generativity decreases to 0 and again co-extension, this time with the rank r+1, etc. The corank r* increases with +1 for each extension, hence it increases a lot. Over time r becomes relatively low compared to r*. There are finally a hand of independent techniques in world of independent working systems. We can recognise here the technical dynamics of automotive or aeronautic industry.
- Conversely, the “*co-extension-driven*” regime favors co-extensions. We have then a symmetrical situation: a hand of independent systems and many independent techniques – but among them there are “generic” techniques that re-organize around themselves the working systems. We recognize here the technical dynamics of semiconductor industry.

Note that this logic of alternating extension and co-extension in the design of techniques could help clarify the open-ended debate in the analysis of the dynamics of techniques and inventions: is the source of technological novelty related to new combinations or is it related to a pure origination, with few antecedents to originate the new technological pathway? What are the relative roles of combinations and originations? (for a recent contribution to this debate see: (Strumsky and Lobo 2015)). Our model provides some elements of answers:
1- we should necessary find both types
2- in an “extension oriented” regime (ie a regime where the design favors the design of working systems), one should record a clear dominance of combination. Actually, one can show that in an extreme extension-driven regime (all extensions are designed before a new co-extension is done), x co-extensions enable approximately x^2 extensions. 1000 co-extensions would enable 1000000 extensions.
These hypotheses are perfectly confirmed by the most recent quantitative study of patents (Strumsky and Lobo 2015): this study identifies the category of “origination” patents (that correspond to co-extensions) and categories for combination (that correspond to extensions). It shows an overwhelming dominance of combination and the relationship between cumulated co-extensions patent and cumulated extension patents shows a power relationship, with a power between 1 and 2, meaning that

patents correspond to an extension driven regime, still not the most extreme one (not all extensions are patented before one co-extension appears!).

5 MAIN FINDINGS AND CONCLUSION

In this paper we studied the design of techniques with systemic impact – such as generic techniques or, more generally, a technique that changes the interdependences between existing techniques. In particular we wanted to build a *model* that would account for the systemic effect, in terms of independence, genericity and generativity and that would account for the design of techniques with specific systemic impact. We built a model of the design of knowledge structure by combining one design theory (C-K theory) and a model of independences in knowledge structures (matroid theory). This C-K/Ma model brought us following findings:

1) Model features:

b) F1: C-K/Ma offers a computational model of the process of designing a technique with systemic impact. C-K/Ma characterizes main operations (extension, co-extension, etc.) and their "quantified" effect (rank, co-rank, generativity, genericity).

c) Since artefacts appear as "combinations of techniques", it is often said that design is combination and one is tempted to assimilate design to combinatorics, which prevents to understand the generative logic of design combinations. This work clarifies different meanings and forms of combinations, from non-generative combinatorics to generative one:
   a. Classical combinatorics: identify one circuit that already exist in a matroid
   b. Deletion-extraction: design one artifact from existing techniques, by deleting some interdependencies with other techniques
   c. Extension: create a new edge (new technique) to create a new working system, taking into account all interdependencies. The new edge is actually dependent on the previously known one, its design is driven by a concept of working system.
   d. Co-extension: create a new edge that increases the dimension of the whole set of techniques, ie open new extension possibilities, ie create new opportunities to invent new combinations.

d) One related finding: this model helps to overcome false intuitions. In particular we show that the usual "lego like" combinative models can be assimilated to one specific matroid, U(1,n) – this matroid is not extendable and the family is not stable by co-extension so that the lego-like "combinative" model actually prevents to analyse the endogenous dynamics of technical systems.

e) Another related findings: in engineering design, the "ideal" system (following Suh's first axiom) is actually a U(n-1,n) matroid and this system prevents the design of generic technique. This means that if one represents an engineering system with independent techniques, then it is impossible to represent generic technique in such a model.

2) Lessons for the design of generic techniques:

a) F2: C-K/Ma offers an efficient model to account for some phenomena assocated to generic technology design. In particular C-K/MA helps understanding the phenomenology of generic techniques such as:
   a. a generic technique –like the cinetic parallelogram in steam engine- does not seem to add functional value to the system (the steam engine and the machine tools were known and did not require a cinetic parallelogram to work!) but is finally in every machines (because this is the key technique to combine the two previously independent working systems) ;
   b. a generic technique couples and decouples, it creates a "modular" relations between working systems – it reorganizes technical system in a flexible way.
   c. A generic technique appears first as just coupling two systems without adding direct value – but it has a strong "indirect value", in the sens that many new techniques can then be added to the newly coupled working systems.

b) F3: C-K/Ma offers a guide for the design of technologies with systemic impact, based on generativity and genericity criteria. C-K/Ma uncovers a large variety of design strategies for pervasive techniques and provides criteria to evaluate how these techniques change the generativity and the genericity of the newly created technical system.

c) Finally we also show that, in an iterative perspective, an endogenous dynamics would necessarily rely on extension and co-extension (ie the design of working system and the design of generic techniques) (otherwise the dynamics would stop). It seems that historical cases (like the industries of the 20th century) tend to rely mainly on extension processes – with rare co-extensions, ie few generic techniques- still it is possible to describe theoretical trajectories with regular coextensions. The model confirms also quantitative empirical works on the source of technological novelty in patents.

Finally, relying on advanced models of design theory and matroid, this paper strengthens the study of the endogenous dynamics of technical systems and opens new ways. Beyond the phenomena and the strategy, the model might also be useful to study economics and organizational issues. In particular the logic of "indirect" value in the design of generic technology actually raises interesting institutional issues: are companies – more interested for direct value- able to design generic technologies with high indirect value? Who could be the new actors in charge of these designs? To study the design of generic techniques we integrated in the model the logics of cohesion and interdependences between techniques – this might now lead us to shed new light on the logics of cohesion and interdependences in economics and society.

REFERENCES

Arthur WB (2009) *The Nature of Technology. What it is and how it evolves.* . Free Press, New York

Baldwin CY, Clark KB (2000) *Design Rules, volume 1: The power of modularity*. The MIT Press, Cambridge, MA, USA

Braha D, Reich Y (2003) Topologial structures for modelling engineering design processes. *Research in Engineering Design* 14 (4):185-199.

Bresnahan TF, Trajtenberg M (1995) General Purpose Technologies: Engines of Growth? *Journal of Econometrics* 65 (1):83-108.

Crapo HH, Rota GC (1970) *On the foundations of combinatorial theory: combinatorial geometries*. MIT Press, Cambridge, MA

Dickinson HW (1936) *Matthew Boulton*. Réédition de 1999 edn. TEE Publishing, Warwickshire, England

Dosi G, Freeman C, Nelson R, Silverberg G, Soete L (eds) (1988) *Technical Change and Economic Theory*. London

Gawer A (ed) (2009) *Platforms, Markets and Innovation*. Edward Elgar, Cheltenham, UK and Northampton, MA

Gilles B (1986) *The History of Techniques* vol 2 vols. Vol. 1: Techniques and civilizations; vol. 2: Techniques and sciences. Gordon and Breach Science Publishers, New York

Hatchuel A, Le Masson P, Reich Y, Weil B (2011) A systematic approach of design theories using generativeness and robustness. In: *International Conference on Engineering Design, ICED'11*, Copenhagen, Technical University of Denmark, 2011. p 12

Hatchuel A, Weil B, Le Masson P (2013) Towards an ontology of design: lessons from C-K Design theory and Forcing. *Research in Engineering Design* 24 (2):147-163.

Hooge S, Kokshagina O, Le Masson P, Levillain K, Weil B, Fabreguette V, Popiolek N (2014) Designing generic technologies in Energy Research: learning from CEA technologies for double unknown management. Paper presented at the *European Academy of Management*, Valencia, Spain,

Kokshagina O, Le Masson P, Weil B (2013a) How design theories enable the design of generic technologies: notion of generic concepts and Genericity building operators Paper presented at the *International Conference on Engineering Design, ICED'13*, Séoul, Korea,

Kokshagina O, Le Masson P, Weil B, Cogez P (2012) Platform emergence in double unknown: common challenge strategy. Paper presented at the *R&D Management Conference*, Grenoble, France,

Kokshagina O, Le Masson P, Weil B, Cogez P (2013b) Platform emergence in double unknown (technology, markets): common unknown strategy. In: Çetindamar D, Daim T, Başoğlu N, Beyhan B (eds) *Strategic planning decisions in the high tech industry*. Springer, London, pp 90-120

Le Masson P, Hatchuel A, Weil B (2013) Teaching at Bauhaus: improving design capacities of creative people? From modular to generic creativity in desing-driven innovation. Paper presented at the *10th European Academy of Design*, Gothenburg,

Le Masson P, Weil B (2013) Design theories as languages for the unknown: insights from the German roots of systematic design (1840-1960). *Research in Engineering Design* 24 (2):105-126.

Le Masson P, Weil B, Kokshagina O (2015) A new perspective for risk management: a study of the design of generic technology with a matroid model in C-K theory. In: Taura T (ed) *Principia Designae - Pre-Design, Design, and Post-Design - Social Motive for the Highly Advanced Technological Society*. Springer, Tokyo, pp 199-219

Neel DL, Neudauer NA (2009) Matroids you have known. *Mathematics magazine* 82 (1):26-41.

Oxley J (2011) *Matroid Theory*. Oxford Graduate Texts in Mathematics, 2nd edition edn. Oxford University Press,

Reich Y (1995) A Critical Review of General Design Theory. *Research in Engineering Design* 7:1-18.

Saviotti PP, Metcalfe JS (1991) *Evolutionary Theories of Economic and Technological Change*. Harwood Academic Publishers, Newark, N.J.

Shai O, Reich Y (2004a) Infused Design: I Theory. *Research in Engineering Design* 15 (2):93-107.

Shai O, Reich Y (2004b) Infused Design: II Practice. *Research in Engineering Design* 15 (2):108-121.

Simondon G (1958) *Du mode d'existence des objets techniques*. L'invention philosophique, 3ème édition de 1989 edn. Aubier,

Strumsky D, Lobo J (2015) Identifying the sources of technological novelty in the process of invention. *Research Policy* 44 (8):1445-1461.

Suh NP (2001) *Axiomatic Design: advances and applications*. Oxford University Press, Oxford

Thurston RH (1878) *A History of the Growth of the Steam Engine*. Appleton, New York

Ulrich KT, Eppinger SD (2008) *Product Design and Development*. 4th edn. Mc Graw Hill,

Whitney H (1935) On the Abstract Properties of Linear Dependance. *American Journal of Mathematics* 57 (3):509-533.

Yoshikawa H (1981) General Design Theory and a CAD System. In: Sata T, Warman E (eds) *Man-Machine Communication in CAD/CAM, proceedings of the IFIP WG5.2-5.3 Working Conference 1980 (Tokyo)*. Amsterdam, North-Holland, pp 35-57